

# Thesaurus-based Efficient Example Retrieval by Generating Retrieval Queries from Similarities\*

Takehito Utsuro<sup>†</sup> Kiyotaka Uchimoto<sup>†</sup> Mitsutaka Matsumoto<sup>†</sup> Makoto Nagao<sup>‡</sup>

<sup>†</sup>Graduate School of Information Science    <sup>‡</sup>Dept. of Electrical Engineering

Nara Institute of Science and Technology    Kyoto University

## Abstract

In example-based NLP, the problem of computational cost of example retrieval is severe, since the retrieval time increases in proportion to the number of examples in the database. This paper proposes a novel example retrieval method for avoiding full retrieval of examples. The proposed method has the following three features, 1) it generates *retrieval queries* from similarities, 2) efficient example retrieval through the tree structure of a thesaurus, 3) binary search along subsumption ordering of retrieval queries. Example retrieval time drastically decreases with the method.

## 1 Introduction

Since a model of machine translation (MT) called *Translation by Analogy* was first proposed in Nagao (1984), much work has been undertaken in example-based NLP (e.g. Sato and Nagao (1990) and Kurohashi and Nagao (1993)). The basic idea of example-based approach to NLP is to accomplish some task in NLP by imitating a similar previous example, instead of using rules written by human writers. Major processing steps of example-based approach are: 1) collect examples and the results of performing the task in a database, 2) given an input, retrieve similar examples from the database, 3) adapt the results of the similar examples to the current input and obtain the output.

Compared with the traditional rule-based approach, example-based approach has advantages like: 1) it is easier to maintain the implemented system, since once the system is constructed, the performance can be improved just by adding new examples, 2) finer-grained syntactic and semantic discrimination can be expected just by adopting finer-grained similarity measure between the input and the example.

In almost all the previous frameworks of example-based NLP, it is necessary to calculate similarity values for all the examples in the database in order to find the most similar one, and this is called *full retrieval*. Usually, the computational cost of example retrieval causes a severe problem, because the retrieval time increases in proportion to the number of examples in the database.

This paper proposes a novel method for avoiding full retrieval. The proposed method, which we call *query generation retrieval*, has the following three features,

1) it generates retrieval queries from similarities, 2) efficient example retrieval through the tree structure of a thesaurus, 3) binary search along subsumption ordering of retrieval queries. In this paper, we focus on retrieval of example surface case structures of Japanese sentences. The similarity value between the input and the example is calculated using existing hand-compiled thesaurus. In the following sections, the similarity measure of surface case structures is defined in section 2, then the framework of query generation retrieval is described in section 3.

## 2 Similarity of Surface Case Structures

As a similarity measure of surface case structures, we basically use the similarity measure in Kurohashi and Nagao (1993). Since the authors' similarity measure is intended for calculating similarity between the input surface case structure and a case frame with example nouns, we adjust it to the similarity between two surface case structures. The following describes the data structure of surface case structures and the thesaurus, and gives the definition of the similarity measure.

### 2.1 Data Structure

#### 2.1.1 Surface Case Structure

In general, surface case structure of a Japanese sentence can be represented in feature-structure-like notation as below:

$$\left[ \text{pred: } V, p_1: \begin{bmatrix} \text{pred: } N_1 \\ \text{sem: } Sem_1 \end{bmatrix}, \dots, p_n: \begin{bmatrix} \text{pred: } N_n \\ \text{sem: } Sem_n \end{bmatrix} \right]$$

In this notation,  $V$  is the verb,  $p_1, \dots, p_n$  are the Japanese surface case markers,  $N_1, \dots, N_n$  are case element nouns, and  $Sem_1, \dots, Sem_n$  are the semantic categories of each case element in a thesaurus.

In our task of retrieval of example surface case structures, the input and the examples to be retrieved have to have the same verb. Besides, the similarity value between the input and the example is dependent only on each semantic category. Thus, in this paper, we define the *surface case structure*  $e$  of a sentence as the set of pairs  $\langle p, Sem \rangle$  where  $p$  is a surface case marker and  $Sem$  is the leaf semantic category of the case element noun:<sup>1</sup>

$$e = \{ \langle p_1, Sem_1 \rangle, \dots, \langle p_n, Sem_n \rangle \}$$

<sup>1</sup>In the remainder of this paper, for brevity's sake, we regard nouns as unambiguous and assume that a noun has only one leaf semantic category in the thesaurus, although nouns can be ambiguous and have more than one semantic category in the current implementation.

\*The authors would like to thank Prof. Y. Matsumoto of Nara Institute of Science and Technology, Dr. Y. Den and Dr. E. Sumita of ATR, and Mr. M. Shimbo of Kyoto University, for valuable comments on the draft of the paper.

### 2.1.2 Thesaurus

A thesaurus of nouns is regarded as a tree in which each node represents a semantic category. We define a *thesaurus* of nouns as a rooted directed tree  $\langle SC, E_t \rangle$  where  $SC$  is the set of semantic categories and  $E_t \subseteq SC \times SC$  is the set of directed edges:

$$E_t = \left\{ (Sem_i, Sem_j) \mid Sem_i, Sem_j \in SC \right\}$$

A noun has one (or possibly more) leaf semantic category in the thesaurus. At present we use an on-line thesaurus called *Bunrui Goi Hyou* (BGH) (NLRJ, 1964). BGH has a six-layered abstraction hierarchy and more than 60,000 Japanese words are assigned to the leaves.

## 2.2 Similarity Measure

### 2.2.1 Similarity of Semantic Categories

Before we define the similarity of surface case structures, first we define the similarity of semantic categories in the thesaurus. We define the *similarity*  $sim_s(Sem_1, Sem_2)$  of two semantic categories  $Sem_1$  and  $Sem_2$  as a monotonically increasing function of the most specific common layer  $mscl(Sem_1, Sem_2)$  of  $Sem_1$  and  $Sem_2$  as below:

$mscl$	1	2	3	4	5	6	exact match
$sim_s$	undef	5	7	8	9	10	11

### 2.2.2 Similarity of Surface Case Structures

First, we assume that the similarity measure of surface case structures have to satisfy the following requirements. The similarity should become greater if 1) the number of the corresponding cases becomes greater, or 2) the corresponding case element nouns become more similar, or 3) the ratio of the number of the corresponding cases to the number of the cases in each surface case structure becomes greater.

When calculating the similarity of two surface case structures  $e_1$  and  $e_2$ , first  $e_1$  and  $e_2$  are matched and the set of pairs of the corresponding cases,  $M(e_1, e_2)$ , is constructed. A case  $\langle p_{1i}, Sem_{1i} \rangle$  of  $e_1$  corresponds to a case  $\langle p_{2j}, Sem_{2j} \rangle$  of  $e_2$  only when the surface case markers  $p_{1i}$  and  $p_{2j}$  are the same and  $sim_s(Sem_{1i}, Sem_{2j})$  is defined.<sup>2</sup> Let  $sim_{ps}(m)$  be the similarity of a pair  $m$  of corresponding cases, then the *similarity*  $sim_e(e_1, e_2)$  of the two surface case structures  $e_1$  and  $e_2$  is defined as below:

$$sim_e(e_1, e_2) = \frac{\sum_{m \in M} sim_{ps}(m)}{|M|} \times \sqrt{\frac{|M|}{|e_1|}} \times \sqrt{\frac{|M|}{|e_2|}}$$

where  $|M|$  is the number of the corresponding cases, and  $|e_1|$  and  $|e_2|$  are the number of cases in  $e_1$  and  $e_2$  respectively. The first factor satisfies requirement 1),

<sup>2</sup>In Japanese, there exist several topic-marking post-positional particles such as “*wa*” and “*mo*”, and cases marked by those topic-marking post-positional particles could correspond to cases marked by case-marking post-positional particles such as “*ga-NOM*” and “*wo-ACC*”. Although this paper considers case-marking post-positional particles only, the implemented system can appropriately calculate the similarity of surface case structures in which topic-marking post-positional particles appear.

and the second satisfies 2). The third and the fourth satisfy 3).

For example, the similarity of the surface case structures  $e_1$  and  $e_2$  of Example 1 and 2 is calculated as follows.

#### Example 1

彼 - が<sup>s</sup>      本 - を      買<sup>う</sup>  
 kare - ga    hon - wo    kau  
 he - NOM    book - ACC    buy  
 (He buys a book.)

#### Example 2

彼 - が<sup>s</sup>      息子 - に      ノート - を      買<sup>う</sup>  
 kare - ga    musuko - ni    nooto - wo    kau  
 he - NOM    son - DAT    notebook - ACC    buy  
 (He buys his son a notebook.)

First, the set of pairs of the corresponding cases,  $M(e_1, e_2)$ , is constructed ( $nb=notebook$ ).

$$M(e_1, e_2) = \left\{ \langle \langle \text{が}(ga), Sem_{he} \rangle, \langle \text{が}(ga), Sem_{he} \rangle \rangle, \right. \\ \left. \langle \langle \text{を}(wo), Sem_{book} \rangle, \langle \text{を}(wo), Sem_{nb} \rangle \rangle \right\}$$

In the case of semantic categories in BGH, the results of the similarity calculation are  $sim_s(Sem_{he}, Sem_{he}) = 11$  and  $sim_s(Sem_{book}, Sem_{nb}) = 9$ . Since  $|M|$ ,  $|e_1|$ , and  $|e_2|$  are 2, 2, and 3 respectively,  $sim_e(e_1, e_2)$  is calculated as follows:

$$sim_e(e_1, e_2) = \sqrt{2} \times \frac{11+9}{2} \times \sqrt{\frac{2}{2}} \times \sqrt{\frac{2}{3}} = 11.55$$

## 3 Query Generation Retrieval

Query generation retrieval has the following three features, 1) it generates retrieval queries from similarities, 2) efficient example retrieval through the tree structure of a thesaurus, 3) binary search along subsumption ordering of retrieval queries. Fig. 1 describes the framework of query generation retrieval.

In query generation retrieval, first, given an input surface case structure, a retrieval query is generated for a certain similarity and then example surface case structures which satisfy the similarity are retrieved from the example database. In order to generate a retrieval query which satisfy the given similarity requirement, it is necessary to enumerate all the possible patterns of surface case structures which satisfy the given similarity requirement. We define *similarity template* which enumerates all the possible patterns of calculating similarity between two surface case structures and collect them in a *similarity table*. The similarity table is referred to when generating retrieval queries from the input surface case structure and a certain similarity.

A retrieval query consists of the number of cases of the example to be retrieved, cases which the example to be retrieved should have, and semantic restrictions of case element nouns. In order to quickly retrieve examples which satisfy a retrieval query, for each surface case marker we build a sub-structure of the whole thesaurus of nouns, which we call *sub-thesaurus*. Examples which satisfy the requirements in a retrieval query are quickly retrieved through the tree structure of those sub-thesauri.

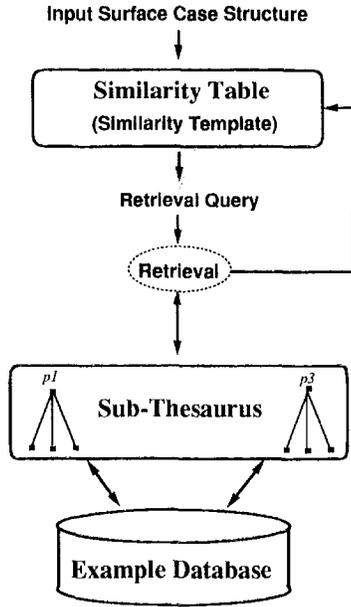


Fig. 1: The Framework of Query Generation Retrieval

In our query generation retrieval, it is necessary to control the retrieval process effectively by providing similarities in a certain order and to retrieve the most similar examples as fast as possible. In this paper, we use *binary search* along subsumption ordering of retrieval queries. It is possible to define a subsumption relation between two retrieval queries. Such subsumption relation of retrieval queries results in the subsumption relation of the sets of retrieved examples. This means that a set of retrieved examples subsumes another set if the retrieval query of the former set subsumes, or in other words, is more general than that of the latter set. With those subsumption relations of retrieval queries and the sets of retrieved examples, it becomes possible to efficiently binary-search the set of examples to be retrieved by the most specific retrieval query.

Sections from 3.1 to 3.3 describe those three features and section 3.4 evaluates the framework.

### 3.1 Retrieval Query Generation from Similarities

#### 3.1.1 Retrieval Query

A *retrieval query*  $q$  is defined as a pair  $\langle l_{db}, csp \rangle$ , where  $l_{db}$  is the number of cases of the example to be retrieved, and  $csp$  is the requirement on cases and semantic restriction of case element nouns, which we call a *case structure pattern*. A case structure pattern is represented as a set of pairs  $\langle p, Sem \rangle$  of a surface case marker  $p$  and a semantic category  $Sem$ , where  $Sem$  is not necessarily a leaf semantic category. It requires that for each element  $\langle p, Sem \rangle$  in  $csp$ , the example to be retrieved has to have a case marked by  $p$  and the case element noun has to satisfy the semantic restriction of the semantic category  $Sem$ .

For example, for the verb “買う (*buy*)”, the following  $q_1$  requires that the example to be retrieved should have three cases, the case element noun of “が<sup>s</sup> (*ga-NOM*)” should be “彼 (*he*)”, and that of the “を (*wo-ACC*)” case should belong to the semantic category of “st”(= “stationery”).  $q_1$  retrieves examples like “彼, が, 息子, に, ノート, を, 買う” (*He buys his son a notebook.*) and “彼, が, 娘, に, 鉛筆, を, 買う” (*He buys his daughter pencils.*).

$$q_1 = \langle 3, \{ \langle \text{が}^s(\text{ga}), Sem_{he} \rangle, \langle \text{を}(\text{wo}), Sem_{st} \rangle \} \rangle$$

#### 3.1.2 Similarity Template

We introduce the notion of *similarity template* in order to enumerate all the possible patterns of calculating the similarity between two surface case structures. In the case of the similarity measure defined in section 2.2.2, a *similarity template* is represented as a 3-tuple:

$$t = \langle l_{in}, l_{db}, CS \rangle \quad (|CS| \leq l_{in}, l_{db} \leq l_{max})$$

where  $l_{in}$  and  $l_{db}$  correspond to the number of cases of the input and of the example respectively, and they are supposed to be less than or equal to the predetermined maximum number  $l_{max}$ .  $CS$  is the multiset of the similarities between corresponding case element nouns. For example, in the case of Example 1 and 2 in section 2.2.2, the result of similarity calculation is represented as a similarity template  $\langle 2, 3, \{11, 9\} \rangle$  (suppose that the former example is the input and the latter is from the example database).

All the possible combinations of  $l_{in}$ ,  $l_{db}$ , and  $CS$  can be enumerated beforehand without any inputs and examples if only the maximum case number  $l_{max}$  is given. Suppose that  $l_{max}$  is 3, the number of possible combinations of  $l_{in}$ ,  $l_{db}$ , and  $CS$  is 203.

#### 3.1.3 Retrieval Query Generation

Similarity templates are collected in the similarity table and referred to when generating retrieval queries from the input and a certain similarity. The following shows how to generate a retrieval query from an input  $e_{in}$  and a similarity template  $t = \langle |e_{in}|, l_{db}, CS \rangle$ .

The retrieval query to be generated is denoted as  $q = \langle l_{db}, csp \rangle$ , where  $l_{db}$  corresponds to the number of cases in the example to be retrieved and is the same as  $l_{db}$  in  $t$ .  $CS$  in  $t$  is the multiset of the similarities between corresponding case element nouns. When constructing the case structure pattern  $csp$  from  $e_{in}$  and  $CS$ , we use an injection to map each similarity  $sim$  in  $CS$  to a case  $\langle p, Sem_{in} \rangle$  in  $e_{in}$ . For each  $\langle p, Sem_{in} \rangle$  to which a similarity  $sim$  is mapped, a case  $\langle p, Sem \rangle$  is collected into  $csp$ , where the semantic category  $Sem$  satisfies  $sim_s(Sem_{in}, Sem) = sim$ .

For example, let the input  $e_{in}$  be the surface case structure of Example 1 and the similarity template  $t$  be  $\langle 2, 3, \{11, 9\} \rangle$ , then there exist two possible injections from  $CS$  into  $e_{in}$  and two retrieval queries are generated as below ( $sim_s(Sem_N, Sem_{N,x}) = x$ ):

$$e_{in} = \{ \langle \text{が}^s(\text{ga}), Sem_{he} \rangle, \langle \text{を}(\text{wo}), Sem_{book} \rangle \}$$

$$q_1 = \langle 3, \{ \langle \text{が}^s(\text{ga}), Sem_{he,11} \rangle, \langle \text{を}(\text{wo}), Sem_{book,9} \rangle \} \rangle$$

$$q_2 = \langle 3, \{ \langle \text{が}^s(\text{ga}), Sem_{he,9} \rangle, \langle \text{を}(\text{wo}), Sem_{book,11} \rangle \} \rangle$$

## 3.2 Efficient Example Retrieval with Sub-Thesaurus

Each element  $\langle p, Sem \rangle$  in a case structure pattern  $csp$  requires that the example surface case structure has to have a case marked by  $p$  and the case element noun has to satisfy the semantic restriction of the semantic category  $Sem$ . Given the example database, it is possible to collect examples which satisfy the requirement  $\langle p, Sem \rangle$  beforehand. For each case marker  $p$ , we collect all those sets of examples. Since all the semantic categories forms the whole thesaurus of nouns, non-empty sets of those collected examples also form a sub-structure of the whole thesaurus of nouns. We call it a *sub-thesaurus* for the case marker  $p$ .

Fig. 2 shows an example of the sub-thesaurus for “を (wo-ACC)” case, supposing that the example database contains Example 1 and 2 in section 2.2.2. The most specific common layer of  $Sem_{book}$  and  $Sem_{nb}$  is the layer 5, and the example set is  $\{Eg1\}$  or  $\{Eg2\}$  at layer 6 and 7(leaf), and  $\{Eg1, Eg2\}$  above layer 6.

Given a requirement  $\langle p, Sem \rangle$  and a sub-thesaurus for  $p$ , examples which satisfy the requirement are quickly retrieved through the tree structure of the sub-thesaurus in constant time. Examples which satisfy all the requirements in  $csp$  are obtained as the intersection of all those sets of retrieved examples.

### The Size of A Sub-Thesaurus

We estimate the size of all the sub-thesauri by the total number of elements in the nodes of those sub-thesauri. Let  $N$  be the total number of examples in the example database,  $d$  be the depth of the whole thesaurus of nouns, and  $l_{max}$  be the maximum case number. A case element noun in an example appears in a leaf node and all of the parent nodes of the leaf in a sub-thesaurus, thus appears  $d$  times. Since the number of case element nouns in an example is at most  $l_{max}$ , the number of case element nouns in the example database is at most  $N \times l_{max}$  and the order of the size of all the sub-thesauri is at most  $N \times l_{max} \times d$ , i.e.,  $O(N)$  ( $l_{max} \times d$  is constant).

## 3.3 Binary Search along Subsumption Ordering of Retrieval Queries

### 3.3.1 Subsumption Relation

A subsumption relation can be defined between two retrieval queries, and results in the subsumption relation of the sets of retrieved examples. For example, in the case of the following two retrieval queries  $q_1$  and  $q_2$ ,  $q_2$  has a requirement on the “に (ni-DAT)” case while  $q_1$  does not, and the requirement on the “が (ga-NOM)” case is more specific in  $q_2$  than in  $q_1$ . Thus,  $q_2$  is more specific than  $q_1$ , or in other words,  $q_1$  subsumes  $q_2$ .

$$q_1 = \left\langle 3, \left\{ \left\langle \text{が} (ga), Sem_{human} \right\rangle, \left\langle \text{を} (wo), Sem_{st} \right\rangle \right\} \right\rangle$$

$$q_2 = \left\langle 3, \left\{ \left\langle \text{が} (ga), Sem_{he} \right\rangle, \left\langle \text{を} (wo), Sem_{st} \right\rangle, \left\langle \text{に} (ni), Sem_{human} \right\rangle \right\} \right\rangle$$

Furthermore,  $q_1$  and  $q_2$  are generated from the similarity templates  $t_1 = \langle 2, 3, \{5, 9\} \rangle$  and  $t_2 = \langle 2, 3, \{11, 9, 5\} \rangle$

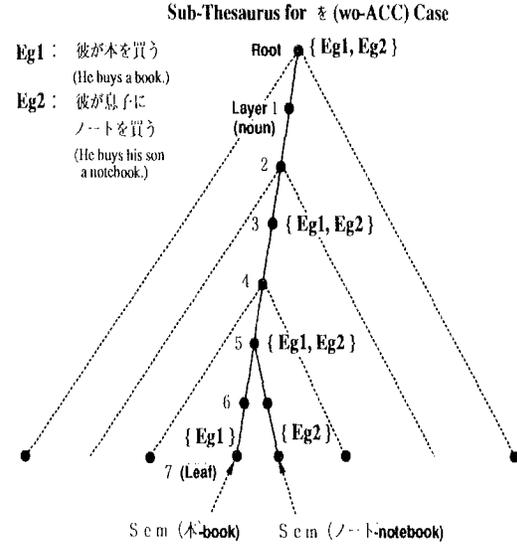


Fig. 2: An Example of Sub-Thesaurus

respectively, and a subsumption relation of similarity templates holds between  $t_1$  and  $t_2$  as well.

### 3.3.2 Similarity Table

With the subsumption relations of retrieval queries and the sets of retrieved examples, it is possible to efficiently binary-search the set of examples to be retrieved by the most specific retrieval query. In the following, we describe how to organize the set of all the similarity templates as a *similarity table* and to realize the process of binary search of the most similar examples.

First, the set of all the similarity templates is divided into a sequence  $T_1, \dots, T_n$  which is totally ordered by the following subsumption relation. Let  $e_{in}$  be the input and  $T_i, T_j (i < j)$  be the sets of similarity templates in the sequence  $T_1, \dots, T_n$ , and  $EG_i, EG_j$  be the sets of examples retrieved by all the similarity templates in  $T_i$  and  $T_j$  respectively. Then,  $T_i$  subsumes  $T_j$  if and only if, 1)  $EG_i$  subsumes  $EG_j$ , and 2) the sets of retrieved examples are totally ordered by similarity, i.e.,  $\forall e_i \in EG_i \forall e_j \in EG_j, sim_e(e_{in}, e_i) < sim_e(e_{in}, e_j)$ .

In the case of the similarity measure defined in section 2.2.2, suppose that  $l_{max}$  is 3, the length of the sequence  $T_1, \dots, T_n$  in the similarity table is 7 when  $l_{in}$  is 1, 9 when  $l_{in}$  is 2, and 11 when  $l_{in}$  is 3.

With this subsumption ordering, the most similar examples are obtained by finding the most specific  $T_i$  with non-empty  $EG_i$  and then finding the most similar examples in  $EG_i$ . Since  $EG_i = \phi$  means  $EG_j = \phi$  for any  $j > i$ , this search process can be efficiently realized by binary-searching the sequence  $T_1, \dots, T_n$ . This sequence  $T_1, \dots, T_n$  can be regarded as a table of similarity templates and is called *similarity table*.

Fig. 3 shows the binary search of the similarity table. Example space is partitioned by the subsumption relation. The most similar examples are found in the innermost non-empty set. This binary search method makes efficient retrieval possible whether the example

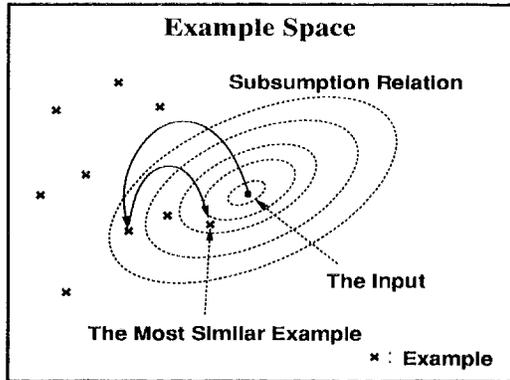


Fig. 3: Binary Search with Subsumption Relation

database contains similar examples or not.

### 3.4 Evaluation

The framework of query generation retrieval consists of three major components, i.e., the *example database*, the *similarity table*, and the *set of sub-thesauri*. Let  $N$  be the size of the example database, the order of the size of the similarity table is independent of  $N$  and that of the set of sub-thesauri is  $O(N)$ . Thus, the total order of the size of the system is  $O(N)$ .

In order to evaluate the computational cost, we plot the computation time (in CPU time), increasing the number of examples  $N$ , and compare the result with a full retrieval program. The example database contains example surface case structures of the Japanese verb “買う (*buy*)” and both programs retrieve the most similar examples from the example database, given an input surface case structure of the same verb “買う (*buy*)”. For the query generation retrieval program, the maximum number  $l_{max}$  of cases was 3. The full retrieval program calculates the similarity between the input and the example for all the examples in the example database, and retrieves the examples with the greatest similarity. Both programs are implemented in SICStus Prolog 2.1 on a SPARC station-10. Fig. 4 illustrates the results. The computation time of the full retrieval program is proportional to  $N$ , while that of the query generation retrieval program is nearly constant. Thus, our query generation retrieval program achieved drastic improvement in decreasing computational cost compared with the full retrieval program.

## 4 Concluding Remarks

This paper proposed a novel example retrieval method in example-based NLP, based on generating retrieval queries from similarities. The proposed method is applicable to any other definitions of similarities in example-based NLP, if the following three requirements are satisfied, 1) the similarity is calculated based on word sense similarities in a thesaurus, 2) the similarity can be defined as a function of a small number of factors, 3) each factor varies over not too many number of discrete values, or otherwise, continuous values can be transformed into not too many discrete values.

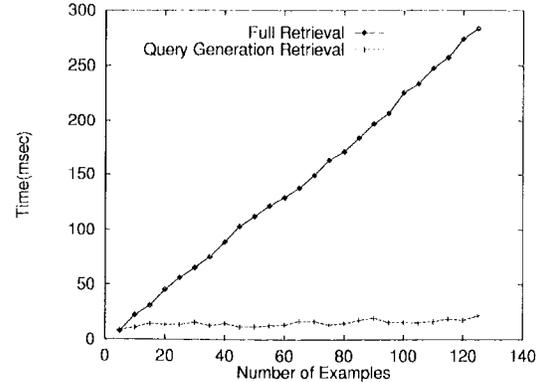


Fig. 4: Time per Number of Examples

In this paper, we restrict ourselves to the cases where examples have the same verb as input. However, similarities of verbs have to be considered and the example database has to contain examples of many verbs. Constructing sub-thesauri might be a problem in those cases, and the solution might be that all the examples are distributed into several sets of sub-thesauri.

In applications like case-based reasoning (CBR) and information retrieval (IR), good examples do not necessarily have the greatest similarity and thus content-based indexing methods are more effective. On the other hand, in example-based NLP, usually similarity measure is strictly defined and only the most similar examples are retrieved. The proposed query generation retrieval fits this feature of example-based NLP.

The idea of generating retrieval queries from similarities is employed also in several other related fields such as translation aid system (Sumita and Tsutsumi, 1988) and CBR (Shimazu, Kitano and Shibata, 1993). Since the tasks of those works are different from ours, their formalizations are different from ours.

## References

- Kurohashi, S. and Nagao, M. (1993). Structural disambiguation in Japanese by evaluating case structures based on examples in case frame dictionary, *Proceedings of the 3rd IWPT*, pp. 111–122.
- Nagao, M. (1984). A framework of a mechanical translation between Japanese and English by analogy principle, in A. Elithorn and R. Banerji (eds), *Artificial and Human Intelligence*, Elsevier Science Publishers, B.V.
- NLRI (National Language Research Institute). (1964). *Word List by Semantic Principles*, Syuei Syuppan. (in Japanese).
- Sato, S. and Nagao, M. (1990). Toward memory-based translation, *Proceedings of the 13th COLING*, Vol. 3, pp. 247–252.
- Shimazu, H., Kitano, H. and Shibata, A. (1993). Retrieving cases from relational data-bases: Another stride towards corporate-wide case-base systems, *Proceedings of the 13th IJCAI*, pp. 909–914.
- Sumita, E. and Tsutsumi, Y. (1988). A translation aid system using flexible text retrieval based on syntax-matching, *Proceedings of the 2nd TMI*.